Conditional Diffusion Models for IR

Yuanzhi Zhu
SP at CVL

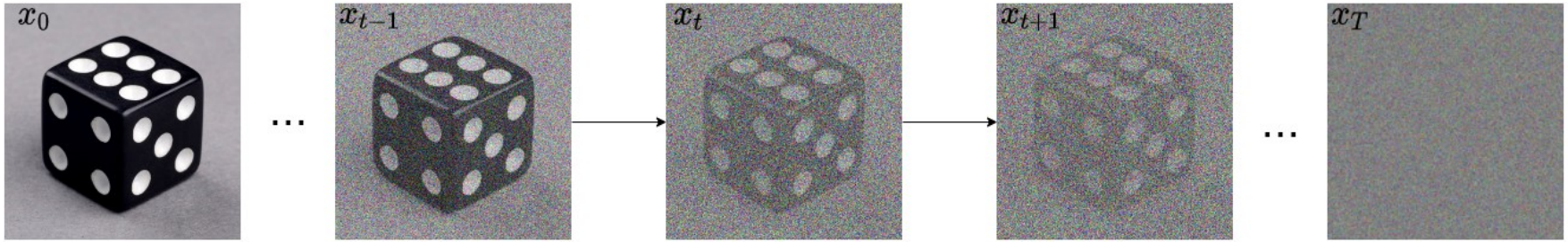
17/Dec/2022

Content

- Preliminaries
- Sampling from the Posterior
- Conditional Models
- Cross-attention Control

DDPM: Denoising Diffusion Probabilistic Models

$$x_0 \sim q(x_0) \longrightarrow q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \longrightarrow q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$



$$p_\theta(x_0) = \int p(x_T) \prod_{i=1}^{T} p_\theta(x_{t-1}|x_t) dx_{1:T} \longleftarrow p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \longleftarrow x_T \sim \mathcal{N}(0, I)$$

DDPM: Denoising Diffusion Probabilistic Models

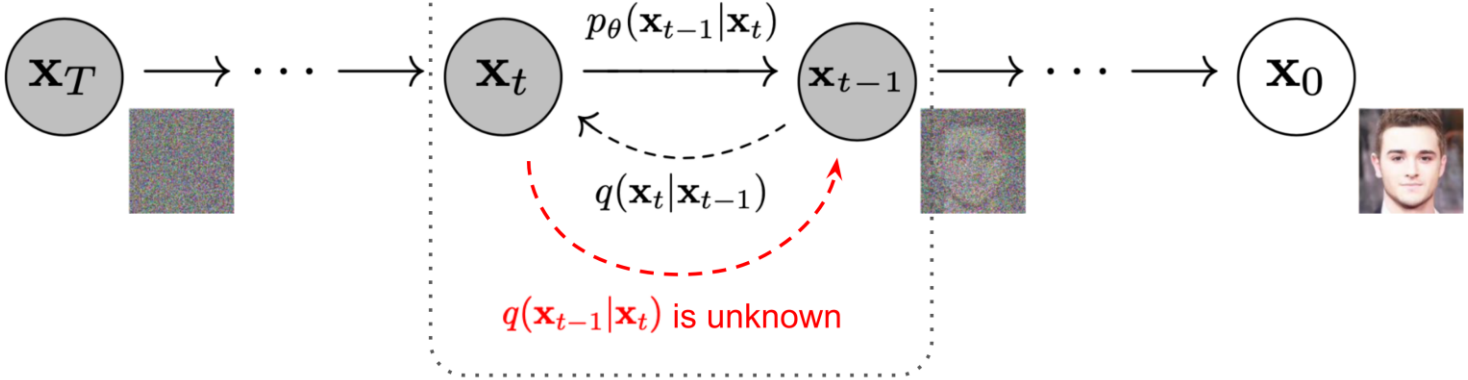
True data dist. : $x_0 \sim q(x_0)$

Forward process: $q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1})$

Reverse process: $p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$

Markov Assumption

Use variational lower bound



DDPM: Denoising Diffusion Probabilistic Models

Forward Diffusion Process

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1})$$

Each Step

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \underbrace{\sqrt{1-\beta_t}x_{t-1}}_{\text{norm invariant}}, \beta_t\mathbf{I}) \quad \text{or} \quad x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}z_{t-1}$$

variance schedule β_t controls the diffusion processing

For arbitrary *t*

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}) \qquad \alpha_t = 1-\beta_t \text{ and } \bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i$$

$$x_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t} \cdot z_t$$

DDPM: Denoising Diffusion Probabilistic Models

Reverse Diffusion Process

if β_t is small enough, $q(x_{t-1}|x_t)$ will also be Gaussian

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \quad p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t,t), \boldsymbol{\Sigma}_\theta(x_t,t))$$

Reverse when condition on x_0

$$\begin{aligned}
&q(x_{t-1}|x_t, x_0) \\
&= q(x_t|x_{t-1}, x_0)\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} \\
&\propto \exp\Big(-\frac{1}{2}\big(\frac{(x_t-\sqrt{\alpha_t}x_{t-1})^2}{\beta_t} + \frac{(x_{t-1}-\sqrt{\bar{\alpha}_{t-1}}x_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t-\sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t}\big)\Big) \\
&= \exp\Big(-\frac{1}{2}\big((\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}})x_{t-1}^2 - (\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0)x_{t-1} + C(x_t,x_0)\big)\Big) \\
&\longrightarrow q(x_{t-1}|x_t, x_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I})
\end{aligned}$$

$$\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 = \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\Big) \qquad \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t$$

z_t is the noise between x_t and x_0

DDPM: Denoising Diffusion Probabilistic Models

Negative Log Likelihood to Variational Lower Bound

$$
\begin{aligned}
-\log p_\theta(\mathbf{x}_0) &\leq -\log p_\theta(\mathbf{x}_0) + D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0)\|p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)) \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{\mathbf{x}_{1:T}\sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})/p_\theta(\mathbf{x}_0)}\right] \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_q\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} + \log p_\theta(\mathbf{x}_0)\right] \\
&= \mathbb{E}_q\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \\
\text{Let } L_{\mathrm{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \geq -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0)
\end{aligned}
$$

DDPM: Denoising Diffusion Probabilistic Models

Parameterization for Training Loss

$$L_{\text{VLB}} = \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] = \mathbb{E}_q\left[\log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right]$$

$$= \mathbb{E}_q\left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right]$$

Known

$$= \mathbb{E}_q[\underbrace{D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_t} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]$$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \boldsymbol{\Sigma}_\theta(x_t, t)) \quad \begin{cases} \boldsymbol{\mu}_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta(x_t, t)) \\ \boldsymbol{\Sigma}_\theta(x_t, t) = \sigma_t^2 \mathbf{I} \end{cases}$$

Model The Noise(Residual)

$$L_t^{\text{simple}} = \mathbb{E}_{x_0, z_t}\left[\|z_t - z_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z_t, t)\|^2\right] \qquad x_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t} \cdot z_t$$

DDPM: Denoising Diffusion Probabilistic Models

Nonetheless, it is just another parameterization of $p_\theta(x_{t-1}|x_t)$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \boldsymbol{\Sigma}_\theta(x_t, t))$$

$$\begin{cases} \boldsymbol{\mu}_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta(x_t, t)) \\ \boldsymbol{\Sigma}_\theta(x_t, t) = \sigma_t^2 \mathbf{I} \end{cases}$$

two options† $\sigma^2 = \begin{cases} \beta_t \\ \frac{1-\hat{\alpha}_{t-1}}{1-\hat{\alpha}_t}\beta_t \end{cases}$

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

† Covariance has analytical optimal form (Estimating the Optimal Covariance with Imperfect Mean in Diffusion Probabilistic Models)

DDIM: Denoising Diffusion Implicit Models*

Variational Inference for *Non-Markovian* Forward Processes

$$-\log p_\theta(\mathbf{x}_0) \leq -\log p_\theta(\mathbf{x}_0) + D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0)\|p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0))$$

Model this directly

$$q_\sigma(x_{1:T}|x_0) = q_\sigma(x_T|x_0)\prod_{t=2}^{T} q_\sigma(x_{t-1}|x_t, x_0)$$

Reverse Process: deterministic given x_t, x_0

$$x_{t-1} = \sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}}\cdot\epsilon_{t-1} = \sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_t + \sigma_t\epsilon$$

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}(\sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\mathbf{I})$$

Forward: still Gaussian (non-Markovian)

$$q_\sigma(x_t|x_{t-1}, x_0) = \frac{q_\sigma(x_{t-1}|x_t, x_0)q_\sigma(x_t|x_0)}{q_\sigma(x_{t-1}|x_0)}$$

[2010.02502] Denoising Diffusion Implicit Models (arxiv.org)

α_t in DDIM is $\bar{\alpha}_t$ in DDPM.

DDIM: Denoising Diffusion Implicit Models

Given: noisy observation x_t

Model difference between x_0 and x_t

Prediction of the corresponding $\hat{x}_0(x_t) = f_\theta^{(t)}(x_t) = (x_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(x_t))/\sqrt{\alpha_t}$

$$p_\theta^{(t)}(x_{t-1}|x_t) = \begin{cases} \mathcal{N}(f_\theta^{(1)}(x_1), \sigma_1^2 I) & \text{if } t = 1 \\ q_\sigma(x_{t-1}|x_t, f_\theta^{(t)}(x_t)) & \text{otherwise} \end{cases}$$

Variational Inference Objective (equivalent to objective in DDPM for certain weights)

$$J_\sigma(\epsilon_\theta) := \mathbb{E}_{\boldsymbol{x}_{0:T}\sim q_\sigma(\boldsymbol{x}_{0:T})}[\log q_\sigma(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) - \log p_\theta(\boldsymbol{x}_{0:T})]$$

$$= \mathbb{E}_{\boldsymbol{x}_{0:T}\sim q_\sigma(\boldsymbol{x}_{0:T})}\left[\log q_\sigma(\boldsymbol{x}_T|\boldsymbol{x}_0) + \sum_{t=2}^{T}\log q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) - \sum_{t=1}^{T}\log p_\theta^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) - \log p_\theta(\boldsymbol{x}_T)\right]$$

Surrogate Objective $L_t^{\text{simple}} = \mathbb{E}_{x_0, z_t}\left[\|z_t - z_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z_t, t)\|^2\right]$ Same as in DDPM!

DDIM: Denoising Diffusion Implicit Models

Sampling from Generalized Generative Processes $p_\theta^{(t)}(x_{t-1}|x_t)$

$$x_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \right)}_{\text{predicted } x_0} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t)}_{\text{direction pointing to } x_t} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}}$$

$$\sigma_t := \eta \sqrt{(1-\alpha_{t-1})/(1-\alpha_t)} \sqrt{1-\alpha_t/\alpha_{t-1}}$$

- **DDPM:** $\eta = 1$ (forward process becomes Markovian (different noise schedule from vanilla DDPM))
- **DDIM:** $\eta = 0$ (forward process becomes deterministic)

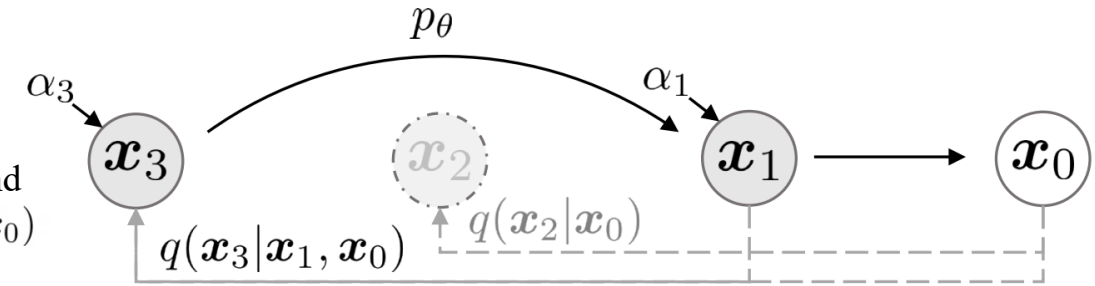
Table 1: CIFAR10 and CelebA image generation measured in FID. $\eta = 1.0$ and $\hat{\sigma}$ are cases of DDPM (although Ho et al. (2020) only considered $T = 1000$ steps, and $S < T$ can be seen as simulating DDPMs trained with S steps), and $\eta = 0.0$ indicates DDIM.

		CIFAR10 (32 × 32)					CelebA (64 × 64)				
	S	10	20	50	100	1000	10	20	50	100	1000
η	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98
	$\hat{\sigma}$	367.43	133.37	32.72	9.99	**3.17**	299.71	183.83	71.71	45.20	**3.26**

DDIM: Denoising Diffusion Implicit Models

Accelerated Generation Processes

Denoising surrogate objective does not depend on the specific forward procedure $q_\sigma(x_{t-1}|x_0)$



Consider the forward process as defined on a subset $\tau = [\tau_1, \tau_2, \ldots, \tau_{\dim(\tau)}] \subset [1, 2, \cdots, T]$

$$q_{\sigma,\tau}(x_{\tau_{i-1}}|x_{\tau_t}, x_0) = \mathcal{N}(x_{\tau_{i-1}}; \sqrt{\bar{\alpha}_{t-1}}x_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2}\frac{x_{\tau_i} - \sqrt{\bar{\alpha}_t}x_0}{\sqrt{1 - \bar{\alpha}_t}}, \sigma_t^2\mathbf{I})$$

The generative process now samples latent variables according to reversed(τ), which we term (sampling) *trajectory*

→ Train a model with arbitrary number forward steps but only sample from some of them in the generative process

SDE-based Generative Models: A Unified Framework*

Forward SDE (data → noise)

$\mathbf{x}(0)$ — $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$ → $\mathbf{x}(T)$

score function $s_\theta(x, \sigma_t)$

$\mathbf{x}(0)$ ← $\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$ — $\mathbf{x}(T)$

Reverse SDE (noise → data)

*[2011.13456] Score-Based Generative Modeling through Stochastic Differential Equations (arxiv.org)

SDE-based Generative Models: A Unified Framework

Training Objective (DSM)

known Gaussian when $f(x,t)$ if affine

$$\theta^* = arg\min \mathbb{E}_{t\sim U(0,T)} \left\{\lambda(t)\mathbb{E}_{x(0)}\mathbb{E}_{x(t)|x(0)}[||s_\theta(x(t),t) - \nabla_{x(t)} \log p_{0t}(x(t)|x(0))||_2^2]\right\}$$

Discretizations

$$dx = f(x,t)dt + g(t)dw$$

SDE Form	Discrete Markov Chain	SDE Expression
Variance Exploding (VE) SDE (NCSN)	$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}$	$\mathrm{d}x = \sqrt{\frac{\mathrm{d}[\sigma^2(t)]}{\mathrm{d}t}}\mathrm{d}w$
Variance Preserving (VP) SDE (DDPM)	$x_i = \sqrt{1-\beta_i}x_{i-1} + \sqrt{\beta_i} z_{i-1}$	$\mathrm{d}x = \frac{1}{2}\beta(t)x\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}w$

SDE-based Generative Models: A Unified Framework

Model: DDPM and SDE point of views

Score in score-based model is affine transformation of predicted noise in DDPM

$$\boldsymbol{x_t} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\cdot\boldsymbol{\varepsilon}$$ Equivalent one step forward

$$\begin{aligned}
\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) &\approx \nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t|\boldsymbol{x}_0) && \text{Denoising score matching} \\
&= -\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{1-\bar{\alpha}_t} && \text{Gaussian assumption} \\
&= -\frac{\boldsymbol{\varepsilon}}{\sqrt{1-\bar{\alpha}_t}} \\
&\approx -\frac{\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}
\end{aligned}$$

Content

- Preliminaries
- Sampling from the Posterior
- Conditional Models
- Cross-attention Control

SDE-based Generative Models: A Unified Framework

Controllable Generation

$$dx = [f(x,t) - g(t)^2 \nabla_x \log p_t(x|y)]dt + g(t)d\bar{w}$$

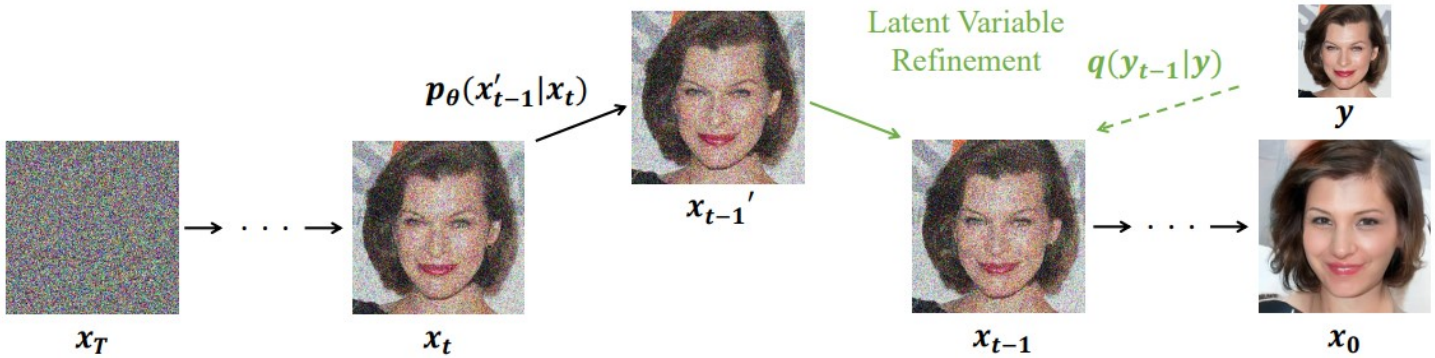
Bayesian

time-dependent classifier(?)

$$dx = \{f(x,t) - g(t)^2 [\nabla_x \log p_t(x) + \nabla_x \log p_t(y|x)]\}dt + g(t)d\bar{w}$$

unconditional model

ILVR: Conditioning Method for DDPM*



Algorithm 1 Iterative Latent Variable Refinement

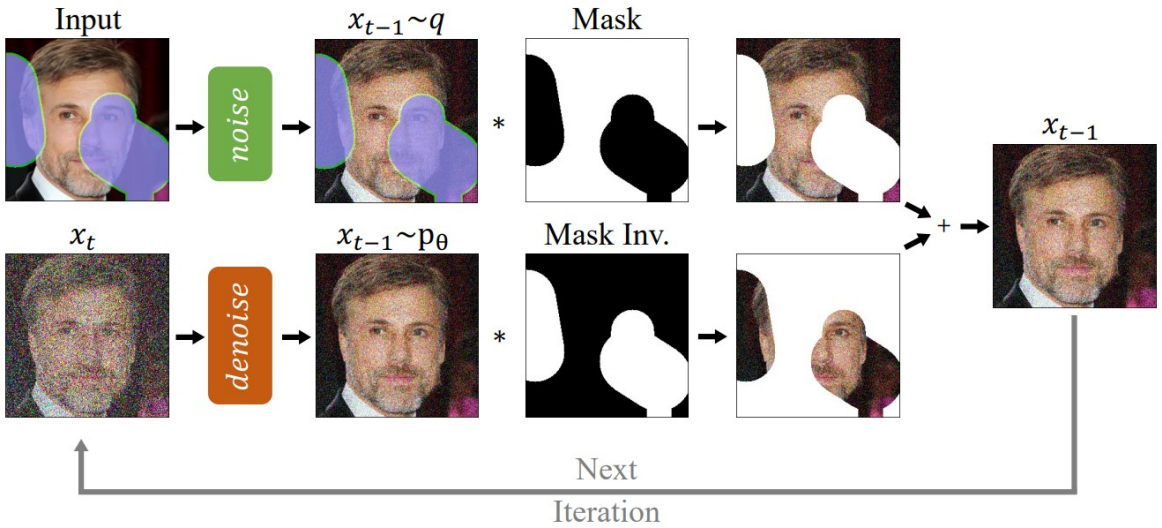
Input: Reference image y
Output: Generated image x
$\phi_N(\cdot)$: low-pass filter with scale N
Sample $x_T \sim N(\mathbf{0}, \mathbf{I})$
for $t = T, ..., 1$ **do**
  $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$
  $x'_{t-1} \sim p_\theta(x'_{t-1}|x_t)$ ▷ unconditional proposal
  $y_{t-1} \sim q(y_{t-1}|y)$ ▷ condition encoding
  $x_{t-1} \leftarrow \phi_N(y_{t-1}) + x'_{t-1} - \phi_N(x'_{t-1})$
end for
return x_0

$$x_{t-1} = x'_{t-1} + a(\phi_N(y_{t-1}) - \phi_N(x'_{t-1}))$$

*[2108.02938] ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models (arxiv.org)

RePaint: Inpainting using Denoising Diffusion Probabilistic Models*

Same idea but different downstream tasks from ILVR



Algorithm 1 Inpainting using our RePaint approach.

1: $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: **for** $u = 1, \dots, U$ **do**
4: $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\epsilon = \mathbf{0}$
5: $x_{t-1}^{\text{known}} = \sqrt{\bar{\alpha}_t} x_0 + (1 - \bar{\alpha}_t)\epsilon$
6: $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
7: $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$
8: $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}$
9: **if** $u < U$ and $t > 1$ **then**
10: $x_t \sim \mathcal{N}(\sqrt{1 - \beta_{t-1}} x_{t-1}, \beta_{t-1}\mathbf{I})$
11: **end if**
12: **end for**
13: **end for**
14: **return** x_0

unconditional

*[2201.09865] RePaint: Inpainting using Denoising Diffusion Probabilistic Models (arxiv.org)

Diffusion Posterior Sampling for General Noisy Inverse Problems*

general forward model $\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}_0) + \boldsymbol{n}, \quad \boldsymbol{y}, \boldsymbol{n} \in \mathbb{R}^n, \boldsymbol{x} \in \mathbb{R}^d$

$$p(\boldsymbol{y}|\boldsymbol{x}_0) = \frac{1}{\sqrt{(2\pi)^n \sigma^{2n}}} \exp\left[-\frac{\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{x}_0)\|_2^2}{2\sigma^2}\right]$$

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$$

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) \simeq \boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t, t) - \rho \nabla_{\boldsymbol{x}_t} \|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0)\|_2^2$$

Algorithm 2 DPS - Gaussian [8]

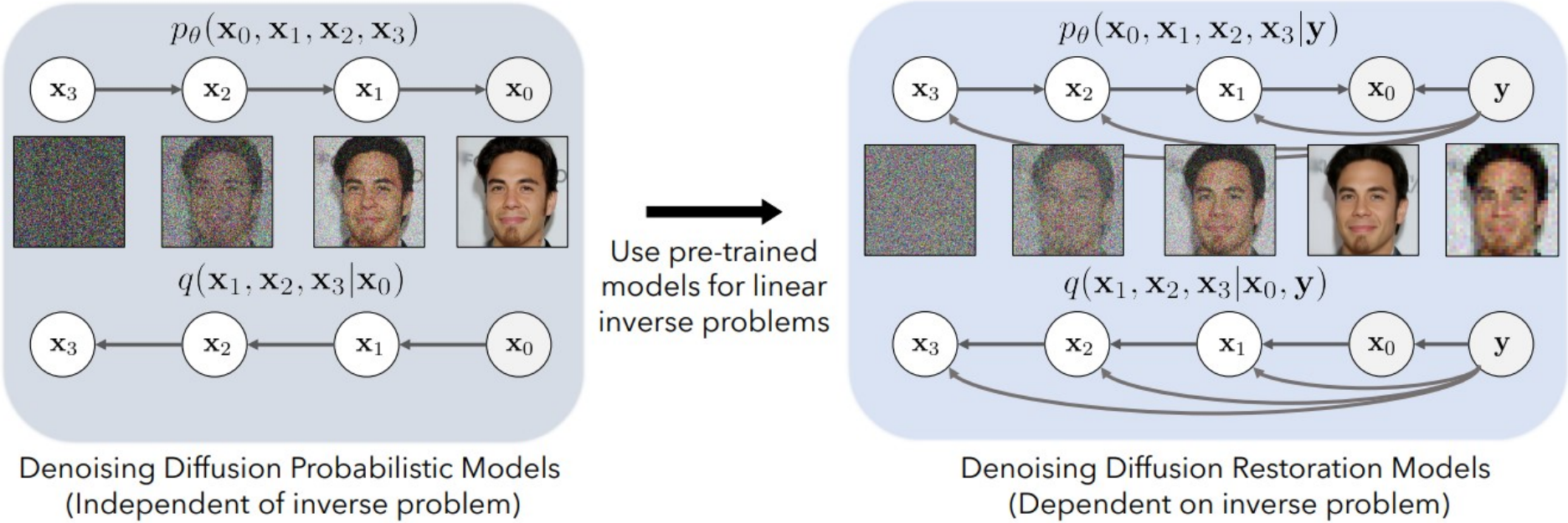
Require: N, y, $\{\zeta_i\}_{i=1}^N$, $\{\tilde{\sigma}_i\}_{i=1^N}$

1: $x_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $i = N-1$ **to** 0 **do**
3: $\quad \hat{s} \leftarrow s_\theta(x_i, i)$
4: $\quad \hat{x}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(x_i + \sqrt{1-\bar{\alpha}_i}\hat{s})$
5: $\quad z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: $\quad x'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i} x_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i} \hat{x}_0 + \tilde{\sigma}_i z$
7: $\quad x_{i-1} \leftarrow x'_{i-1} - \zeta_i \nabla_{x_i} \|y - \mathscr{A}(\hat{x}_0)\|_2^2$
8: **return** x_0

*[2209.14687] Diffusion Posterior Sampling for General Noisy Inverse Problems (arxiv.org)

Denoising Diffusion Restoration Models (DDRM)*

An efficient, unsupervised posterior sampling method



Denoising Diffusion Probabilistic Models
(Independent of inverse problem)

Denoising Diffusion Restoration Models
(Dependent on inverse problem)

*[2201.11793] Denoising Diffusion Restoration Models (arxiv.org)

Denoising Diffusion Restoration Models (DDRM)*

Linear inverse problem $\mathbf{y} = \boldsymbol{H}\mathbf{x} + \mathbf{z} \Rightarrow \bar{\mathbf{y}} = \bar{\mathbf{x}}_0 + \bar{\mathbf{z}}$ $\quad q(\bar{\mathbf{y}}^{(i)}|\mathbf{x}_0) = \mathcal{N}(\bar{\mathbf{x}}_0^{(i)}, \sigma_{\mathbf{y}}^2/s_i^2)$

SVD $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \Rightarrow$ $\bar{\mathbf{x}}_t = V^T\mathbf{x}_t$, $\bar{\mathbf{y}} = \Sigma^\dagger U^T\mathbf{y}$

$$\underbrace{q^{(T)}(\bar{\mathbf{x}}_T^{(i)}|\mathbf{x}_0,\mathbf{y})}_{\text{forward}} = \begin{cases} \mathcal{N}(\bar{\mathbf{y}}^{(i)}, \sigma_T^2 - \frac{\sigma_{\mathbf{y}}^2}{s_i^2}) & \text{if } s_i > 0 \\ \mathcal{N}(\bar{\mathbf{x}}_0^{(i)}, \sigma_T^2) & \text{if } s_i = 0 \end{cases}$$

$$\underbrace{q^{(t)}(\bar{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t+1},\mathbf{x}_0,\mathbf{y})}_{\text{reverse}} = \begin{cases} \mathcal{N}(\bar{\mathbf{x}}_0^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{x}}_{t+1}^{(i)} - \bar{\mathbf{x}}_0^{(i)}}{\sigma_{t+1}}, \eta^2\sigma_t^2) & \text{if } s_i = 0 \\ \mathcal{N}(\bar{\mathbf{x}}_0^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{y}}^{(i)} - \bar{\mathbf{x}}_0^{(i)}}{\sigma_{\mathbf{y}}/s_i}, \eta^2\sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_{\mathbf{y}}}{s_i} \\ \mathcal{N}((1-\eta_b)\bar{\mathbf{x}}_0^{(i)} + \eta_b\bar{\mathbf{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_{\mathbf{y}}^2}{s_i^2}\eta_b^2) & \text{if } \sigma_t \geq \frac{\sigma_{\mathbf{y}}}{s_i} \end{cases}$$

$$p_\theta^{(T)}(\bar{\mathbf{x}}_T^{(i)}|\mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{y}}^{(i)}, \sigma_T^2 - \frac{\sigma_{\mathbf{y}}^2}{s_i^2}) & \text{if } s_i > 0 \\ \mathcal{N}(0, \sigma_T^2) & \text{if } s_i = 0 \end{cases}$$

singular values $s_1 \geq s_2 \geq \ldots \geq s_m$

DDRM $p_\theta^{(t)}(\bar{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t+1},\mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{x}}_{t+1}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{t+1}}, \eta^2\sigma_t^2) & \text{if } s_i = 0 \quad \text{y null-space} \\ \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathbf{y}}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{\mathbf{y}}/s_i}, \eta^2\sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_{\mathbf{y}}}{s_i} \quad \text{final steps} \\ \mathcal{N}((1-\eta_b)\bar{\mathbf{x}}_{\theta,t}^{(i)} + \eta_b\bar{\mathbf{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_{\mathbf{y}}^2}{s_i^2}\eta_b^2) & \text{if } \sigma_t \geq \frac{\sigma_{\mathbf{y}}}{s_i} \quad \text{generative part} \end{cases}$

*[2201.11793] Denoising Diffusion Restoration Models (arxiv.org)

These equations in DDRM are in VE SDE form

Diffusion Model Based Posterior Sampling for Noisy Linear Inverse Problems*

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) \simeq \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t)$$
$$= \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{A}^T \left(\sigma^2 \mathbf{I} + \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t} \mathbf{A}\mathbf{A}^T \right)^{-1} \left(\mathbf{y} - \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{A}\mathbf{x}_t \right)$$

A itself is row-orthogonal

$$[\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t)]_m = \frac{\mathbf{a}_m^T \left(\mathbf{y} - \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{A}\mathbf{x}_t \right)}{\sigma^2 \sqrt{\bar{\alpha}_t} + \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}} \|\mathbf{a}_m\|_2^2}$$

efficient computation via SVD

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) \simeq \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t)$$
$$= \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{V}\mathbf{\Sigma} \left(\sigma^2 \mathbf{I} + \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t} \mathbf{\Sigma}^2 \right)^{-1} \left(\mathbf{U}^T \mathbf{y} - \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{\Sigma}\mathbf{V}^T \mathbf{x}_t \right), \quad (12)$$

Algorithm 1: DMPS: DM based posterior sampling

Input: $\mathbf{y}, \mathbf{A}, \sigma^2, \{\tilde{\sigma}_t\}_{t=1}^T, \lambda$

Initialization: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

for $t = T$ ***to*** 1 **do**

- Draw $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \right) + \tilde{\sigma}_t \mathbf{z}_t$
- Compute $\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t)$ as (12)
- $\mathbf{x}_{t-1} = \mathbf{x}_{t-1} + \lambda \frac{1-\alpha_t}{\sqrt{\alpha_t}} \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t)$

Output: $\mathbf{x}_0$

*[2211.12343] Diffusion Model Based Posterior Sampling for Noisy Linear Inverse Problems (arxiv.org)

Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model*

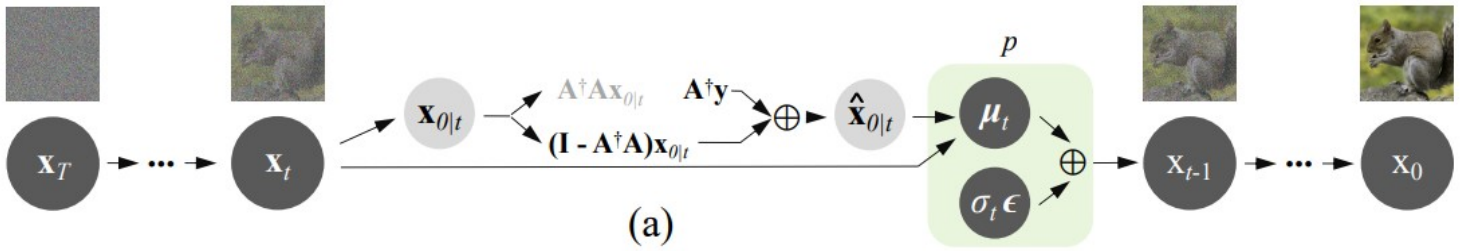
Decouple $\mathbf{x} \equiv \underbrace{\mathbf{A}^\dagger\mathbf{A}\mathbf{x}}_{\text{range-space of } \mathbf{A}} + \underbrace{(\mathbf{I} - \mathbf{A}^\dagger\mathbf{A})\mathbf{x}}_{\text{null-space of } \mathbf{A}}$

$$\textit{Consistency}: \quad \mathbf{A}\hat{\mathbf{x}} \equiv \mathbf{y}, \qquad \textit{Realness}: \quad \hat{\mathbf{x}} \sim q(\mathbf{x})$$

Reconstruction $\hat{\mathbf{x}} = \mathbf{A}^\dagger\mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger\mathbf{A})\bar{\mathbf{x}}$

find a proper $\bar{x}$ that makes the null-space term is in harmony with the range-space term

Diffusion Models $\hat{\mathbf{x}}_{0|t} = \mathbf{A}^\dagger\mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger\mathbf{A})\mathbf{x}_{0|t}$



*[2212.00490] Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model (arxiv.org)

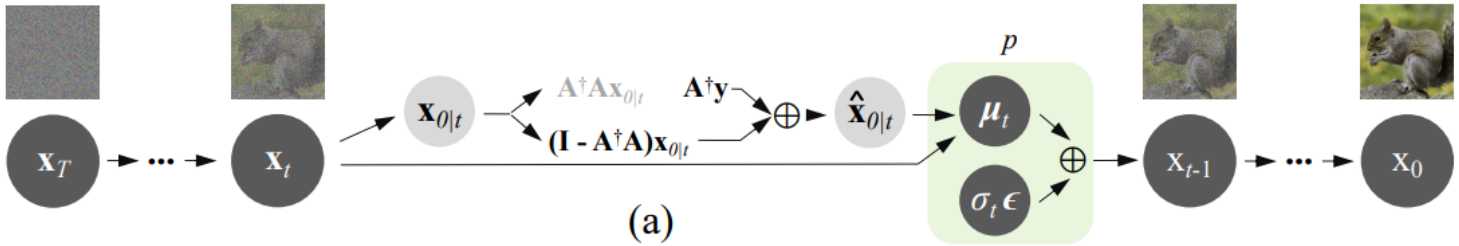
Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model*

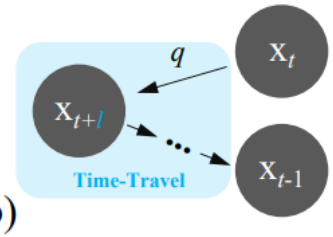
Algorithm 1 Sampling of DDNM

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{x}_{0|t} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\sqrt{1-\bar{\alpha}_t}\right)$
4: $\quad \hat{\mathbf{x}}_{0|t} = \mathbf{A}^\dagger \mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A})\mathbf{x}_{0|t}$
5: $\quad \mathbf{x}_{t-1} \sim p(\mathbf{x}_{t-1}|\mathbf{x}_t, \hat{\mathbf{x}}_{0|t})$
6: **return** $\mathbf{x}_0$

Algorithm 2 Sampling of DDNM+

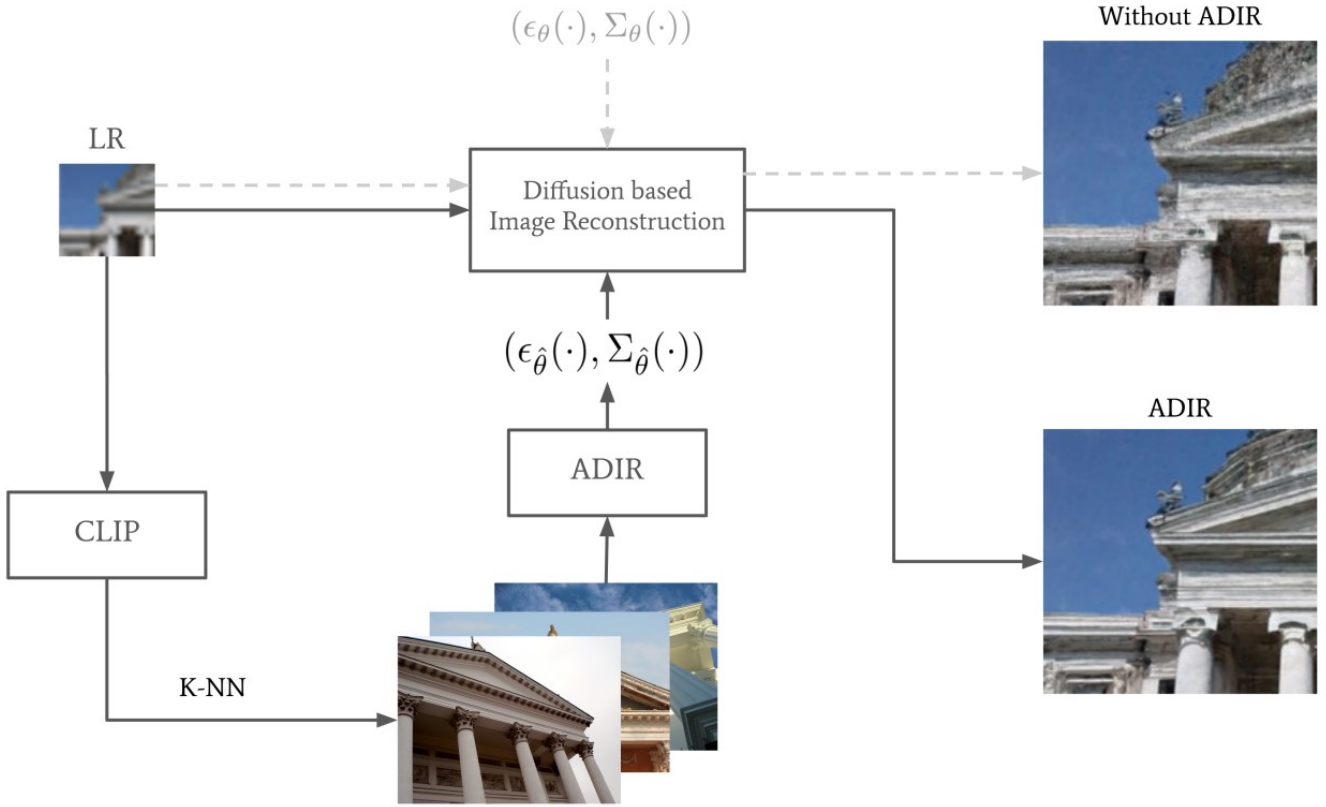
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad L = \min\{T - t, l\}$
4: $\quad \mathbf{x}_{t+L} \sim q(\mathbf{x}_{t+L}|\mathbf{x}_t)$
5: $\quad$ **for** $j = L, \ldots, 0$ **do**
6: $\quad\quad \mathbf{x}_{0|t+j} = \frac{1}{\sqrt{\bar{\alpha}_{t+j}}} \left(\mathbf{x}_{t+j} - \mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x}_{t+j}, t+j)\sqrt{1-\bar{\alpha}_{t+j}}\right)$
7: $\quad\quad \hat{\mathbf{x}}_{0|t+j} = \mathbf{x}_{0|t+j} - \boldsymbol{\Sigma}_{t+j}\mathbf{A}^\dagger(\mathbf{A}\mathbf{x}_{0|t+j} - \mathbf{y})$
8: $\quad\quad \mathbf{x}_{t+j-1} \sim \hat{p}(\mathbf{x}_{t+j-1}|\mathbf{x}_{t+j}, \hat{\mathbf{x}}_{0|t+j})$
9: **return** $\mathbf{x}_0$





*[2212.00490] Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model (arxiv.org)

ADIR: Adaptive Diffusion for Image Reconstruction*



*[2212.03221] ADIR: Adaptive Diffusion for Image Reconstruction (arxiv.org)
ADIR: Adaptive Diffusion for Image Reconstruction (shadyabh.github.io)

ADIR: Adaptive Diffusion for Image Reconstruction*

Algorithm 1 Proposed guided diffusion sampling for image reconstruction given a diffusion model $(\boldsymbol{\epsilon}_\theta(\cdot), \boldsymbol{\Sigma}_\theta(\cdot))$, and a guidance scale s

Require: $(\boldsymbol{\epsilon}_\theta(\cdot), \boldsymbol{\Sigma}_\theta(\cdot))$, $\mathbf{y}$, s

1: $\mathbf{x}_T \leftarrow$ sample from $\mathcal{N}(\mathbf{0}, \mathbf{I}_n)$
2: **for all** t from T to 1 **do**
3: $\quad \hat{\boldsymbol{\epsilon}}, \hat{\boldsymbol{\Sigma}} \leftarrow \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$
4: $\quad \hat{\boldsymbol{\mu}} \leftarrow \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\boldsymbol{\epsilon}})$ } unconditional
5: $\quad \mathbf{y}_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{y} + \sqrt{1-\bar{\alpha}_t}\mathbf{A}\hat{\epsilon}$
6: $\quad \mathbf{g} \leftarrow -2\mathbf{A}^T(\mathbf{A}\hat{\boldsymbol{\mu}} - \mathbf{y}_t)$
7: $\quad \mathbf{x}_{t-1} \leftarrow$ sample from $\mathcal{N}(\hat{\boldsymbol{\mu}} + s\hat{\boldsymbol{\Sigma}}\mathbf{g}, \hat{\boldsymbol{\Sigma}})$
8: **end for**
9: **return** $\mathbf{x}_0$

$$\hat{\boldsymbol{\mu}}_\theta = \boldsymbol{\mu}_\theta + a\boldsymbol{\Sigma}_\theta \nabla_x \log p_\phi(y|\mathbf{x}_{t-1})$$

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y}|\boldsymbol{x}_t) \simeq -\rho \nabla_{\boldsymbol{x}_t} \|\boldsymbol{y} - \mathcal{A}(\boldsymbol{x}_0)\|^2_{\boldsymbol{\Lambda}}, \quad [\boldsymbol{\Lambda}]_{ii} \triangleq 1/2\boldsymbol{y}_j$$

$$\mathbf{g} \approx -2\mathbf{A}^T(\mathbf{A}\mathbf{x}_t - \mathbf{y}_t)|_{\mathbf{x}_t = \boldsymbol{\mu}_\theta}$$

Adaptation scheme:

retrieve K images to fine-tuning the diffusion model

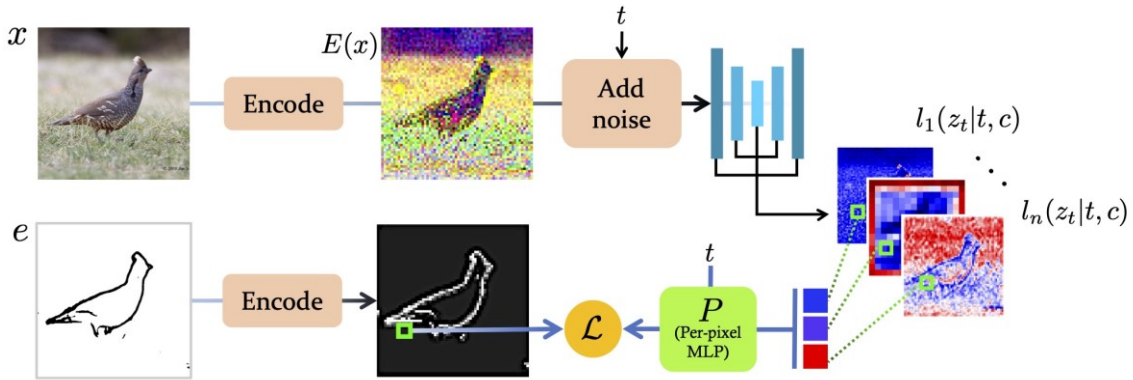
spherical distance between embeddings

$$\{\mathbf{z}_k\}_{k=1}^K = \{\mathbf{z}_1, ..., \mathbf{z}_K | \phi_\xi(\mathbf{z}_1, \mathbf{y}) \leq ... \leq \phi_\xi(\mathbf{z}_K, \mathbf{y}) \leq \phi_\xi(\mathbf{z}, \mathbf{y}), \forall \mathbf{z} \in \mathcal{D}_{\text{IA}} \setminus \{\mathbf{z}_1, ..., \mathbf{z}_K\}\},$$

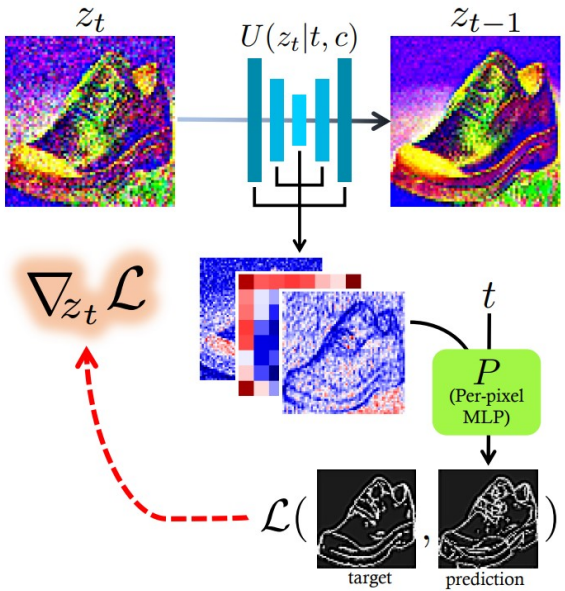
*[2212.03221] ADIR: Adaptive Diffusion for Image Reconstruction (arxiv.org)
ADIR: Adaptive Diffusion for Image Reconstruction (shadyabh.github.io)

Sketch-Guided Text-to-Image Diffusion Models*

(1) Train Per-pixel MLP for sketch-image spatial alignment



(2) Evaluate the anti-gradient as edges-guidance



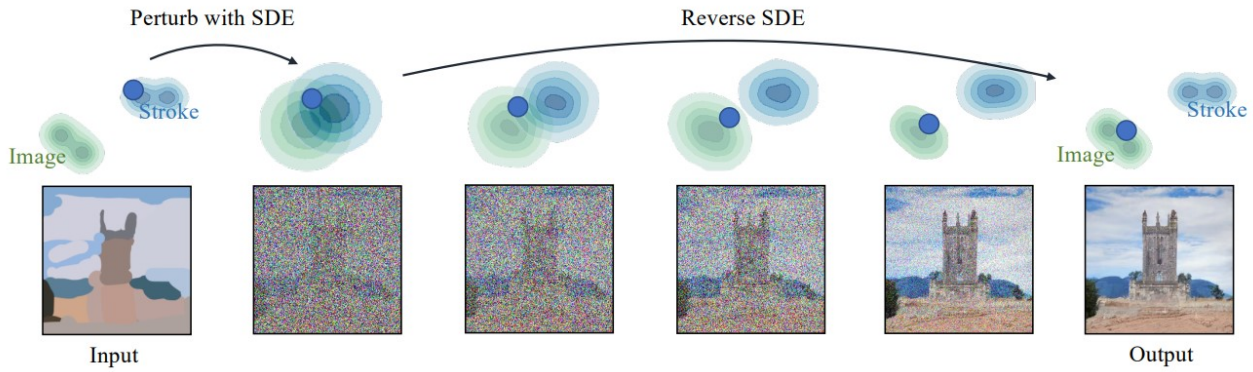
Update similar to DPS: $\tilde{z}_{t-1} = z_{t-1} - \alpha \cdot \nabla_{z_t}\mathcal{L}$

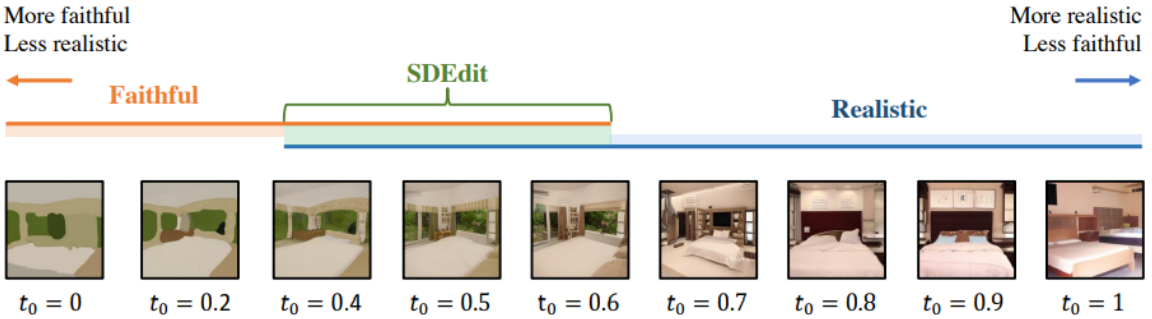
*[2211.13752] Sketch-Guided Text-to-Image Diffusion Models (arxiv.org)
Sketch Guidance (sketch-guided-diffusion.github.io)

Content

- Preliminaries
- Sampling from the Posterior
- Conditional Models
- Cross-attention Control

SDEdit: Guided Image Synthesis and Editing with SDE*

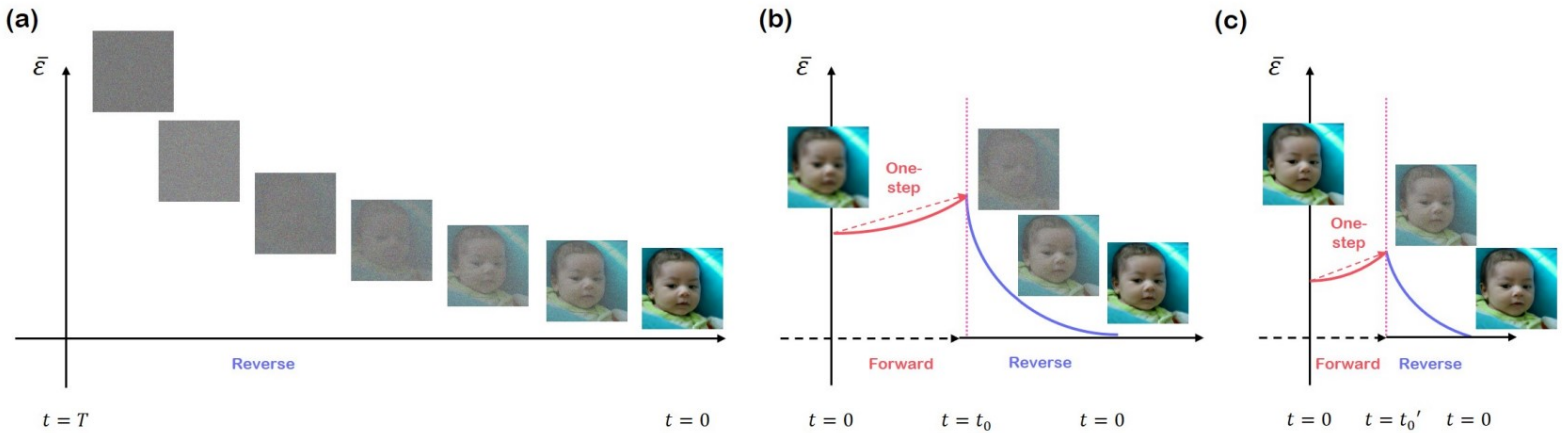




*[2108.01073] SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations (arxiv.org)

Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction*

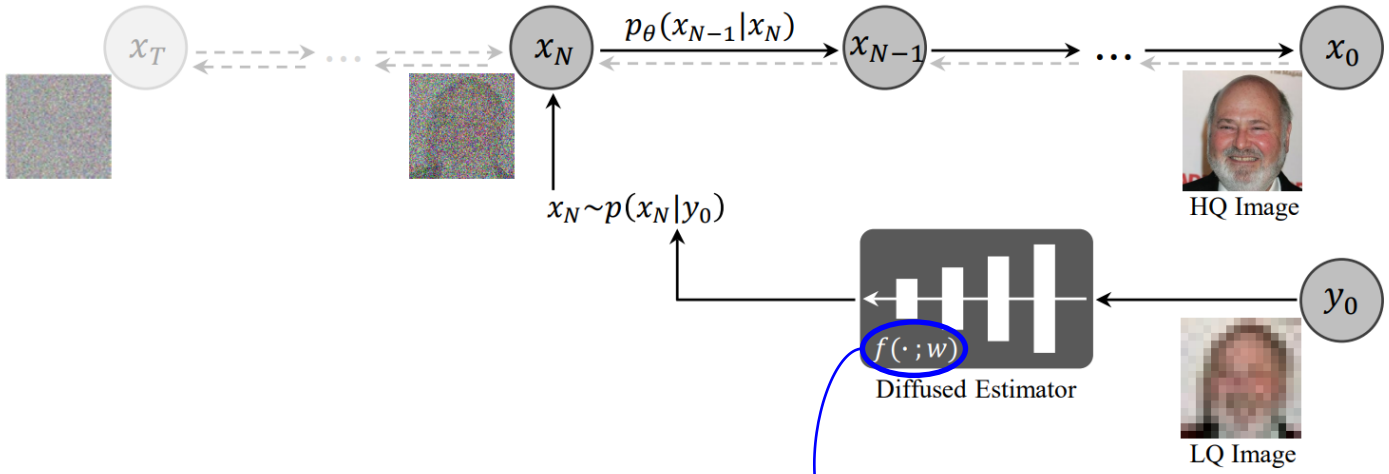
Same idea but different downstream tasks: super-resolution (SR), inpainting, and MRI reconstruction



*[2112.05146] Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction

DifFace: Blind Face Restoration with Diffused Error Contraction*

Same idea but different downstream task: Blind face (easier) restoration



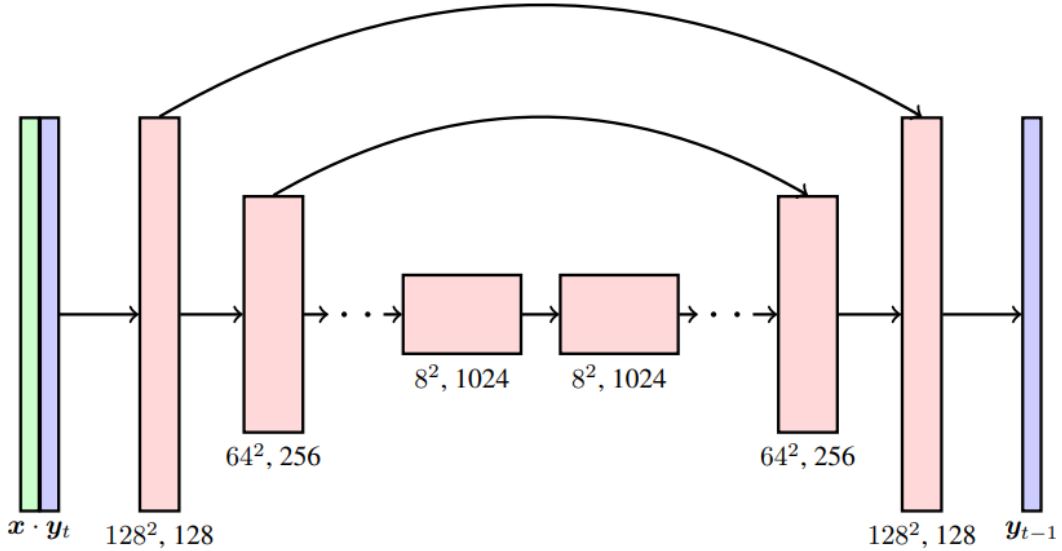
$$\boldsymbol{y} = \left\{ [(\boldsymbol{x} * \boldsymbol{k}_l) \downarrow_s + \boldsymbol{n}_\sigma]_{\text{JPEG}_q} \right\} \uparrow_s$$

two classical network architectures
as the backbone SRCNN and SwinIR

*[2212.06512] DifFace: Blind Face Restoration with Diffused Error Contraction (arxiv.org)

Image Super-Resolution via Iterative Refinement*

The condition is concatenated with y_t along the channel dimension (cascaded)



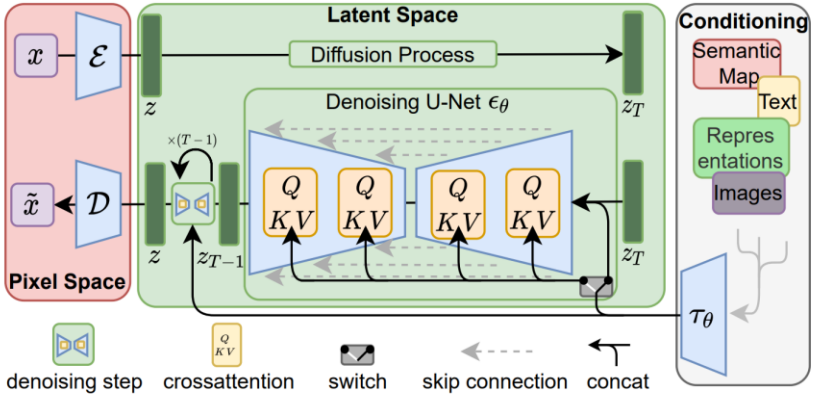
Same author also proposed palette for multi-tasks†, same architecture used for cascaded diffusion‡

*[2104.07636] Image Super-Resolution via Iterative Refinement (arxiv.org)
†[2111.05826] Palette: Image-to-Image Diffusion Models (arxiv.org)
‡[2106.15282] Cascaded Diffusion Models for High Fidelity Image Generation (arxiv.org)

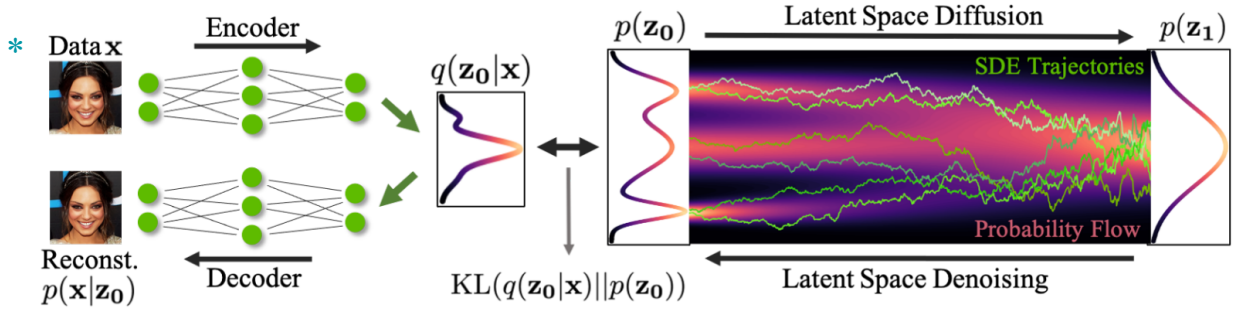
Score-based Generative Modeling in Latent Space*

Faster diffusion in latent space

†

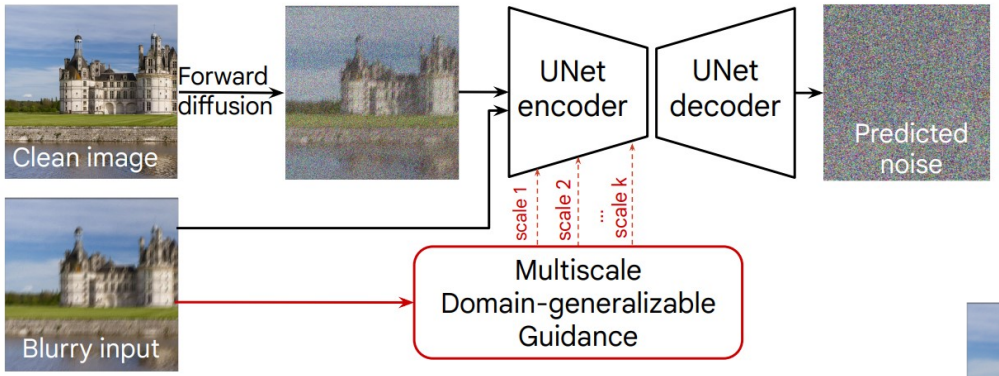


*

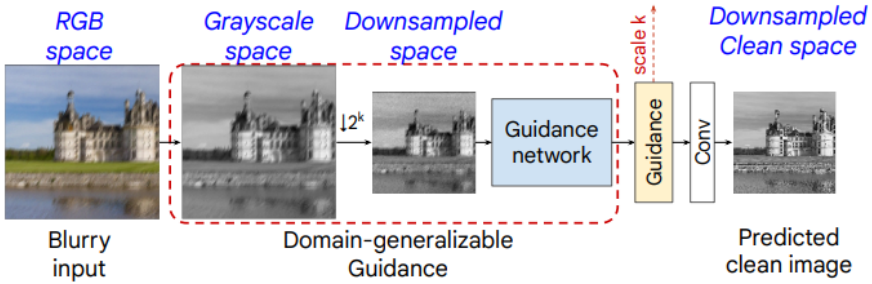


†[2112.10752] High-Resolution Image Synthesis with Latent Diffusion Models (arxiv.org)
*[2106.05931] Score-based Generative Modeling in Latent Space (arxiv.org)

Image Deblurring with Domain Generalizable Diffusion Models*



multiscale domain-generalizable representation that removes domain-specific information while preserving the underlying image structure



*[2212.01789] Image Deblurring with Domain Generalizable Diffusion Models (arxiv.org)

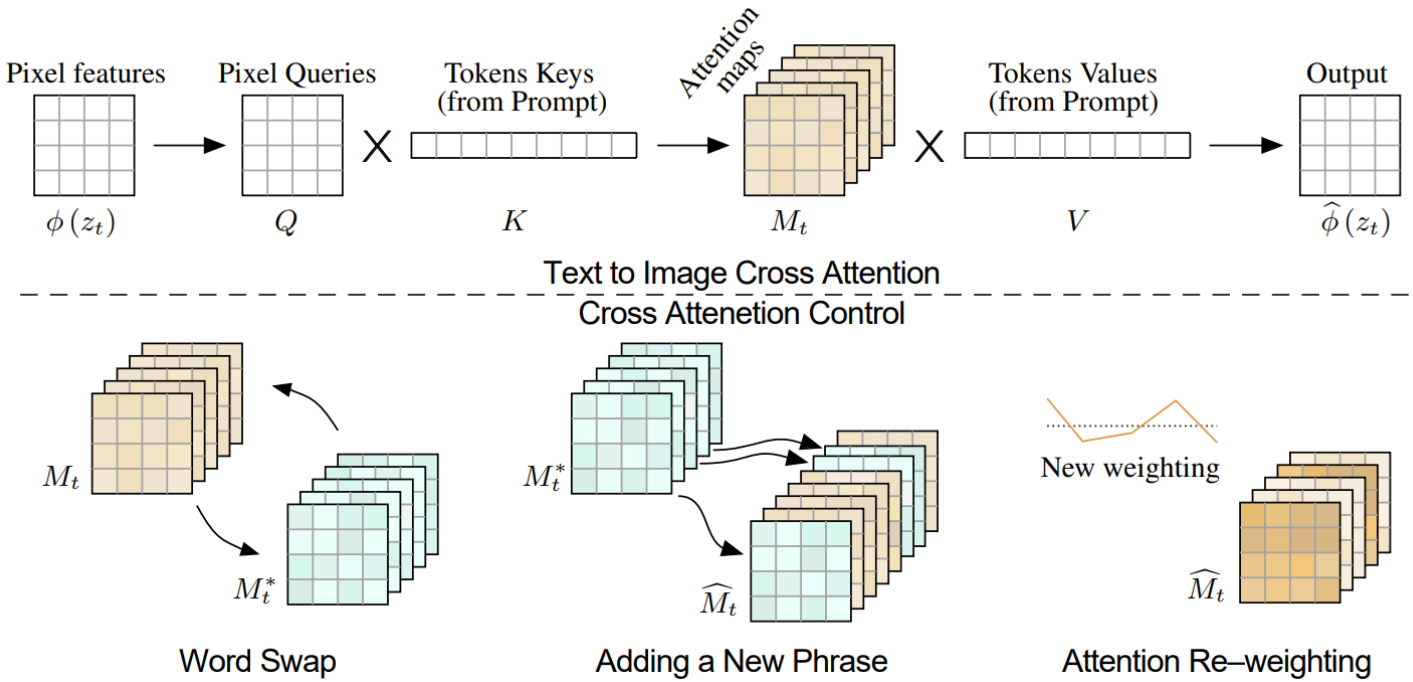
Content

- Preliminaries
- Sampling from the Posterior
- Conditional Models
- Cross-attention Control

Prompt-to-Prompt Image Editing with Cross Attention Control*

$$M = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)$$

we can inject the attention maps M that were obtained from the generation with the original prompt P, into a second generation with the modified prompt P^*



*[2208.01626] Prompt-to-Prompt Image Editing with Cross Attention Control (arxiv.org)

Prompt-to-Prompt Image Editing with Cross Attention Control*

Algorithm 1: Prompt-to-Prompt image editing

Input: A source prompt $\mathcal{P}$, a target prompt $\mathcal{P}^*$, and a random seed s.
Output: A source image x_{src} and an edited image x_{dst}.
$z_T \sim N(0, I)$ a unit Gaussian random variable with random seed s;
$z_T^* \leftarrow z_T$;
for $t = T, T-1, \ldots, 1$ **do**
  $z_{t-1}, M_t \leftarrow DM(z_t, \mathcal{P}, t, s)$;
  $M_t^* \leftarrow DM(z_t^*, \mathcal{P}^*, t, s)$;
  $\widehat{M}_t \leftarrow Edit(M_t, M_t^*, t)$;
  $z_{t-1}^* \leftarrow DM(z_t^*, \mathcal{P}^*, t, s_t)\{M \leftarrow \widehat{M}_t\}$;
end
Return (z_0, z_0^*)

Word Swap

$$Edit(M_t, M_t^*, t) := \begin{cases} M_t^* & \text{if } t < \tau \\ M_t & \text{otherwise} \end{cases}$$

Adding a New Phrase

$$(Edit\,(M_t, M_t^*, t))_{i,j} := \begin{cases} (M_t^*)_{i,j} & \text{if } A(j) = None \\ (M_t)_{i,A(j)} & \text{otherwise.} \end{cases}$$

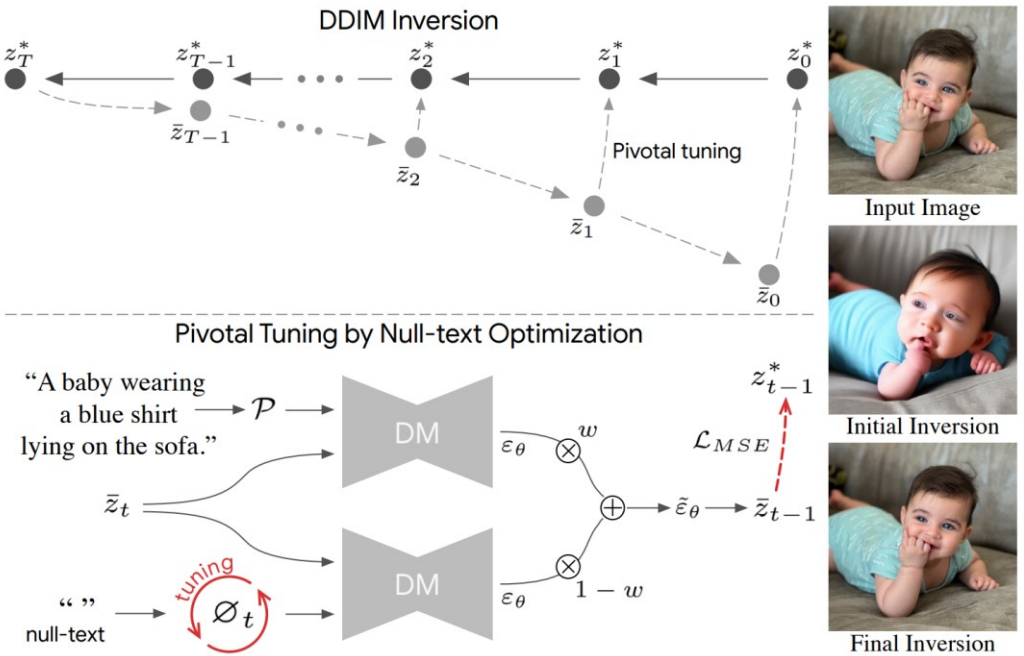
i :pixel value; j :text token

Attention Re–weighting

$$(Edit\,(M_t, M_t^*, t))_{i,j} := \begin{cases} c \cdot (M_t)_{i,j} & \text{if } j = j^* \\ (M_t)_{i,j} & \text{otherwise} \end{cases}$$

*[2208.01626] Prompt-to-Prompt Image Editing with Cross Attention Control (arxiv.org)

Null-text Inversion for Editing Real Images using Guided Diffusion Models*



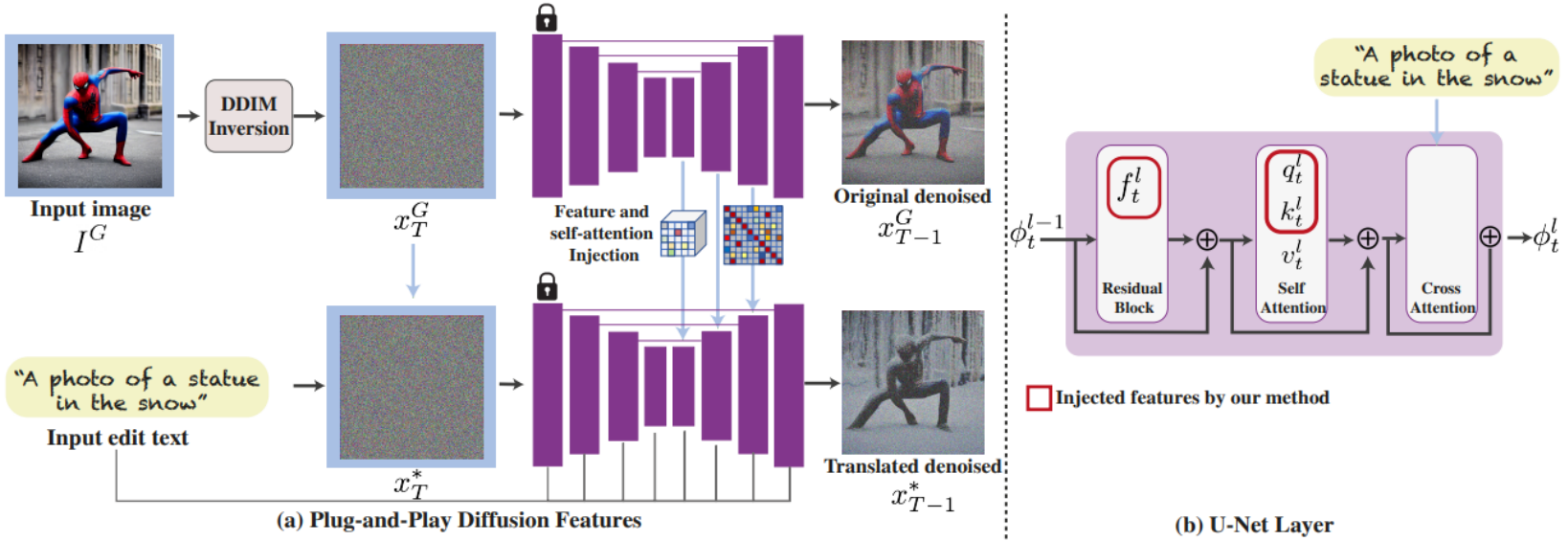
Algorithm 1: Null-text inversion

Input: A source prompt embedding $\mathcal{C} = \psi(\mathcal{P})$ and input image $\mathcal{I}$.
Output: Noise vector z_T and optimized embeddings $\{\varnothing_t\}_{t=1}^{T}$.

Set guidance scale $w = 1$;
Compute the intermediate results $z_T^*, \ldots, z_0^*$ using DDIM inversion over $\mathcal{I}$;
Set guidance scale $w = 7.5$;
Initialize $\bar{z_T} \leftarrow z_T^*$, $\varnothing_T \leftarrow \psi(\text{""})$;
for $t = T, T-1, \ldots, 1$ **do**
  for $j = 0, \ldots, N-1$ **do**
    $\varnothing_t \leftarrow \varnothing_t - \eta \nabla_{\varnothing} \left\| z_{t-1}^* - z_{t-1}(\bar{z}_t, \varnothing_t, \mathcal{C}) \right\|_2^2$
  end
  Set $\bar{z}_{t-1} \leftarrow z_{t-1}(\bar{z}_t, \varnothing_t, \mathcal{C})$, $\varnothing_{t-1} \leftarrow \varnothing_t$;
end
Return $\bar{z_T}, \{\varnothing_t\}_{t=1}^{T}$

*[2211.09794] Null-text Inversion for Editing Real Images using Guided Diffusion Models (arxiv.org)

Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation*



*[2211.12572] Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation (arxiv.org)

Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation*

Algorithm 1 Plug-and-Play Diffusion Features

Inputs:

I^G ▷ real guidance image

P ▷ target text prompt

τ_f, τ_A ▷ injection thresholds

$\boldsymbol{x}_T^G \leftarrow \text{DDIM-inv}(I^G)$

$\boldsymbol{x}_T^* \leftarrow \boldsymbol{x}_T^G$ ▷ Starting from same seed

for $t \leftarrow T \ldots 1$ **do**

$\boldsymbol{z}_{t-1}^G, \boldsymbol{f}_t^4, \left\{\boldsymbol{A}_t^l\right\} \leftarrow \epsilon_\theta\left(\boldsymbol{x}_t^G, \varnothing, t\right)$

$\boldsymbol{x}_{t-1}^G \leftarrow \text{DDIM-samp}\left(\boldsymbol{x}_t^G, \boldsymbol{z}_{t-1}^G\right)$

if $t > \tau_f$ **then** $\boldsymbol{f}_t^{*4} \leftarrow \boldsymbol{f}_t^4$ **else** $\boldsymbol{f}_t^{*4} \leftarrow \varnothing$

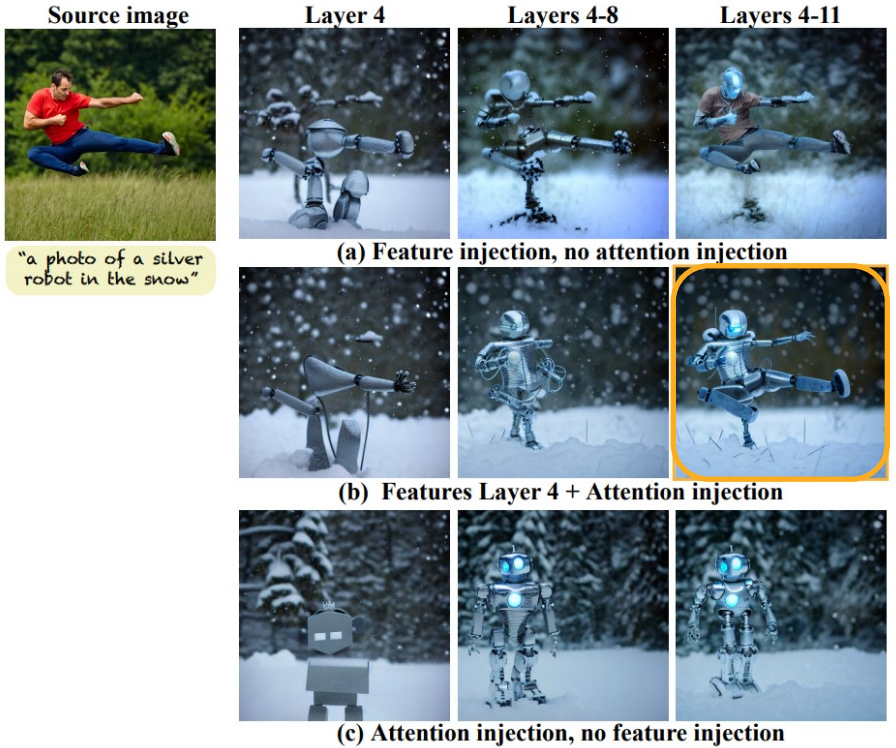
if $t > \tau_A$ **then** $\boldsymbol{A}_t^{*l} \leftarrow \boldsymbol{A}_t^l$ **else** $\boldsymbol{A}_t^{*l} \leftarrow \varnothing$

$\boldsymbol{z}_{t-1}^* \leftarrow \hat{\epsilon}_\theta\left(\boldsymbol{x}_t^*, P, t\,;\, \boldsymbol{f}_t^{*4}, \left\{\boldsymbol{A}_t^{*l}\right\}\right)$

$\boldsymbol{x}_{t-1}^* \leftarrow \text{DDIM-samp}\left(\boldsymbol{x}_t^*, \boldsymbol{z}_{t-1}^*\right)$

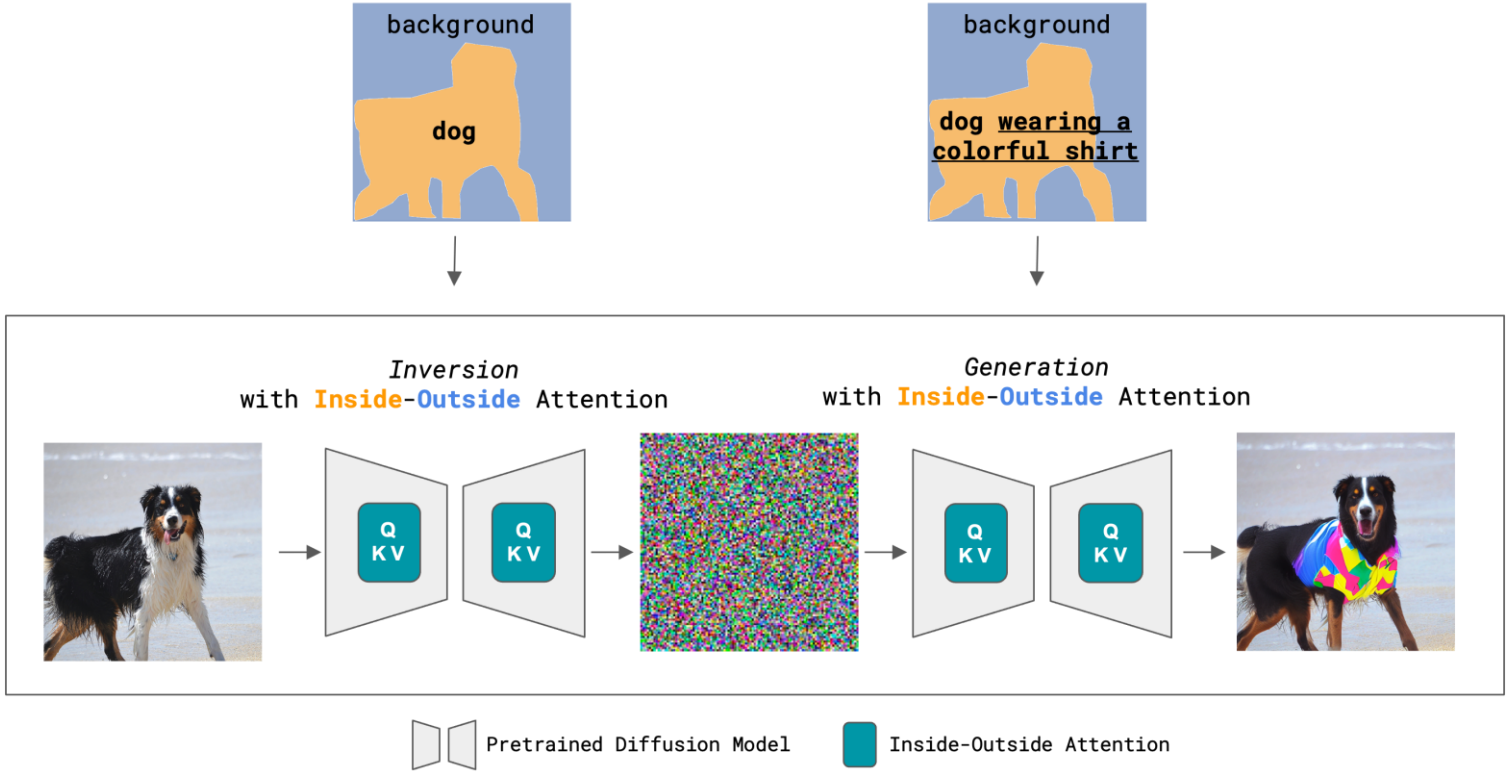
end for

Output: $I^* \leftarrow \boldsymbol{x}_0^*$



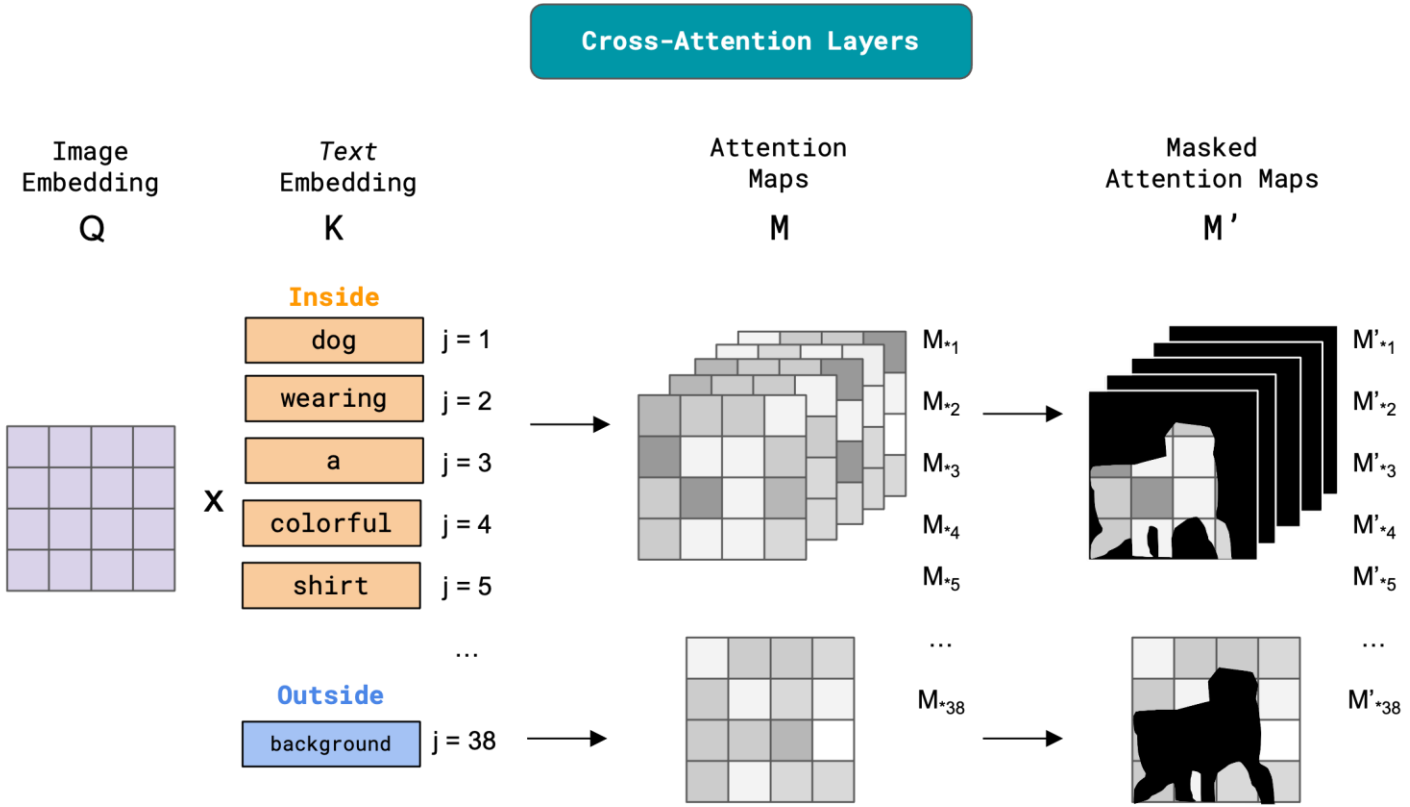
*[2211.12572] Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation (arxiv.org)

Shape Guided Diffusion with Inside-Outside Attention*



*[2212.00210] Shape-Guided Diffusion with Inside-Outside Attention (arxiv.org)
Shape Guided Diffusion with Inside-Outside Attention (shape-guided-diffusion.github.io)

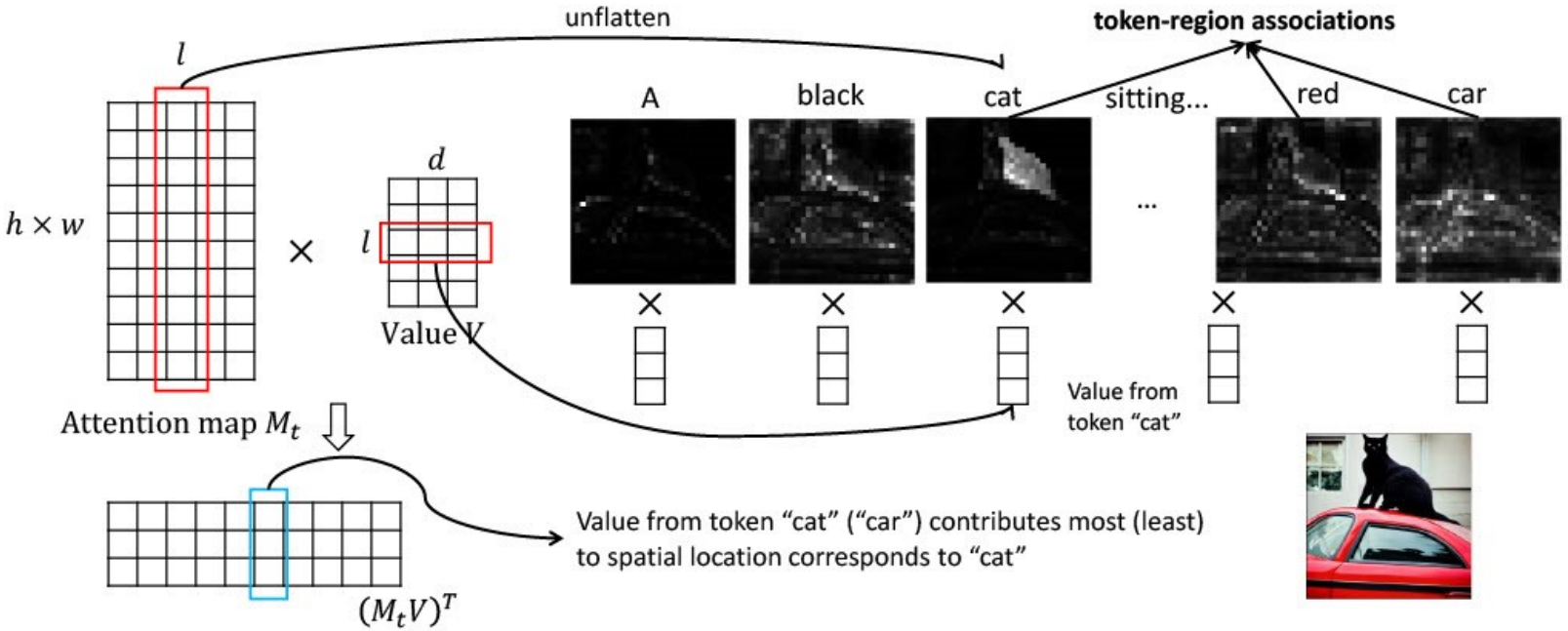
Shape Guided Diffusion with Inside-Outside Attention*



*[2212.00210] Shape-Guided Diffusion with Inside-Outside Attention (arxiv.org)
Shape Guided Diffusion with Inside-Outside Attention (shape-guided-diffusion.github.io)

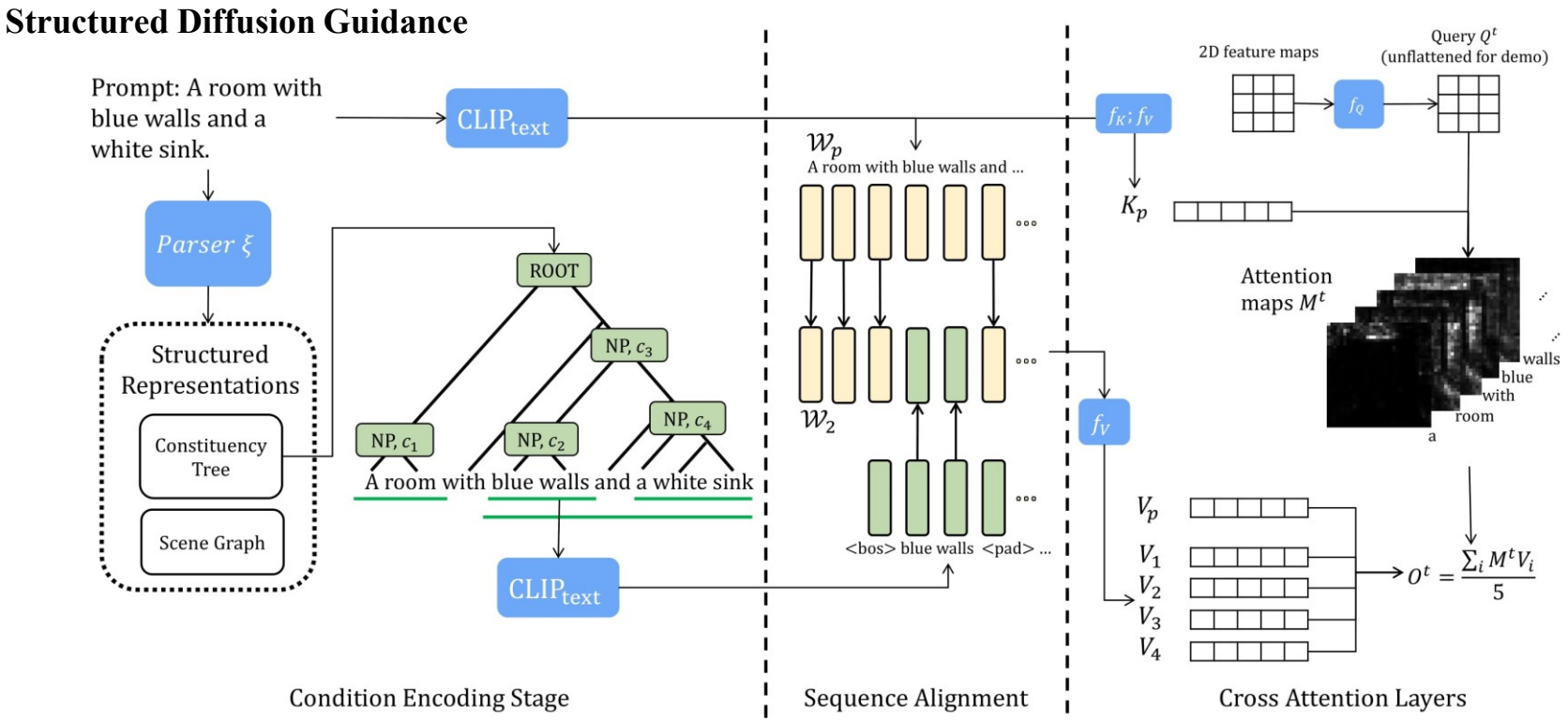
Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis*

Cross-attention control



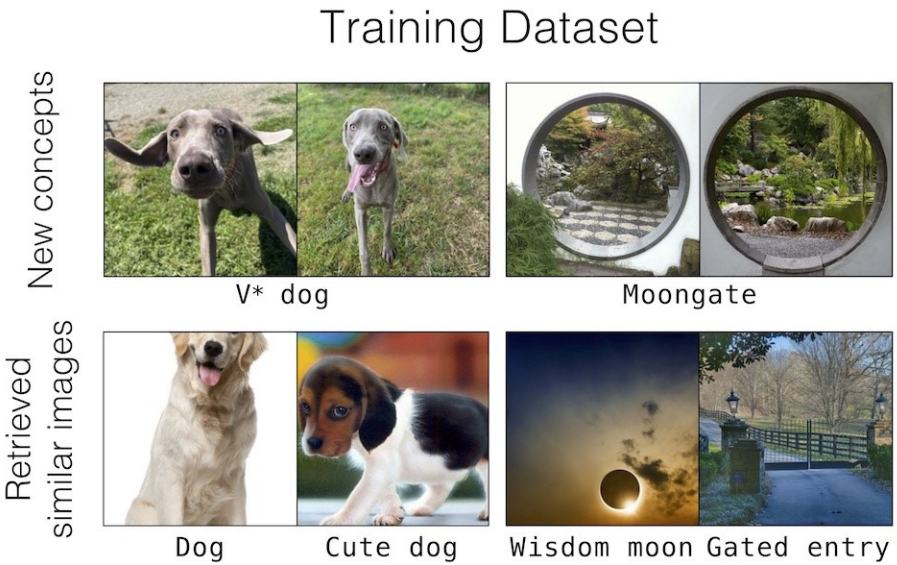
*[2212.05032] Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis (arxiv.org)
StructuredDiffusion (weixi-feng.github.io)

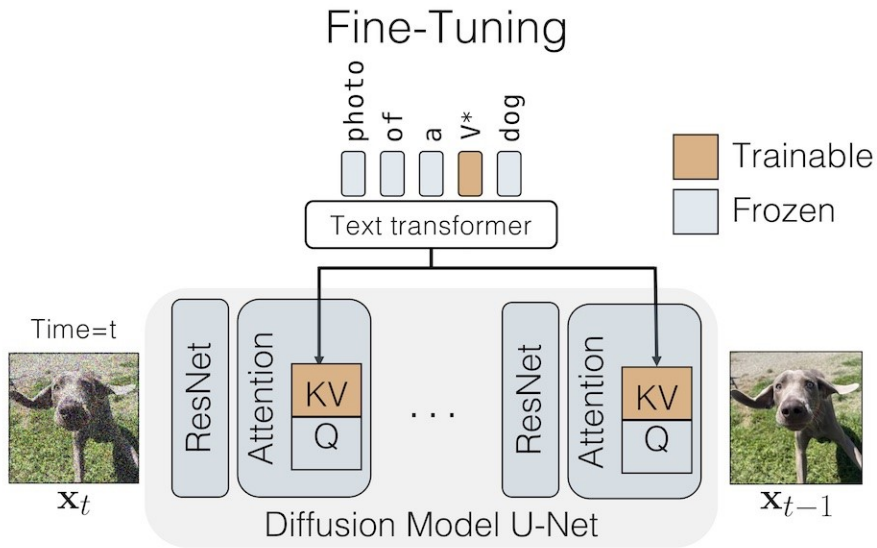
Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis*



*[2212.05032] Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis (arxiv.org)
StructuredDiffusion (weixi-feng.github.io)

Multi-Concept Customization of Text-to-Image Diffusion*





*[2212.04488] Multi-Concept Customization of Text-to-Image Diffusion (arxiv.org)
Multi-Concept Customization of Text-to-Image Diffusion (cmu.edu)